

Disc 9 - OpSem and Lambda Calculus

Thursday, November 4, 2021 11:09 AM

Operational Semantics

2. Using the rules given below, show: $1 + (2 + 3) \Rightarrow 6$

$$\frac{}{n \Rightarrow n} \quad \frac{e_1 \Rightarrow n_1 \quad e_2 \Rightarrow n_2 \quad n_3 \text{ is } n_1 + n_2}{e_1 + e_2 \Rightarrow n_3}$$

4. Using the rules given below, show: $A; \text{let } y = 1 \text{ in let } x = 2 \text{ in } x \Rightarrow 2$

$$\frac{\frac{}{A; n \Rightarrow n} \quad \frac{A(x) = v}{A; x \Rightarrow v} \quad \frac{A; e_1 \Rightarrow v_1 \quad A, x : v_1; e_2 \Rightarrow v_2}{A; \text{let } x = e_1 \text{ in } e_2 \Rightarrow v_2} \quad \frac{A; e_1 \Rightarrow n_1 \quad A; e_2 \Rightarrow n_2 \quad n_3 \text{ is } n_1 + n_2}{A; e_1 + e_2 \Rightarrow n_3}$$

- 5) Recall last week we went over lexing and parsing:

```
type expr =  
  | Int of int  
  | Plus of expr * expr
```

Implement an expression evaluator, that takes an environment closure and an expression, and returns a value after evaluating it.

Key Notes (Taken from OpSem rules, which will be given on the project)

- Integers evaluate to themselves
- Plus works on integers (throw a `TypeError` otherwise)

```
let rec eval_expr env e =
```